texas_merge_postmeeting

Teo Richard

2025-04-19

Setup

Read in datasets (old, new, street names)

Make all NA dates in `old$date_open` be 1993-01-01

Normalizer

```
replacees1 = c('north', 'east', 'west', 'south') # replacing these values with n, e, s, w, s
replacers1 = c('n', 'e', 'w', 's')

# replacing these values (full street names) with nothing
replacees2 = str_to_lower(c(street_names_abbs[[1]],
                 street_names_abbs[[2]], "and", "temporarily closed"))

replacers2 = rep('', length(replacees2))

replacees = append(replacees1, replacees2)
replacers = append(replacers1, replacers2)

replacement = setNames(replacers, replacees) # named vector: values come from replacers, names come fro

# general sort function
sorted = function(col) {

  formatted = str_to_lower(col)
  formatted = str_replace_all(formatted, replacement) # replacing all `replacees` with `replacers`
  formatted = str_replace_all(formatted, "\\b[A-Za-z]\\b", '')
  formatted = str_replace_all(formatted, setNames(c('', ''), c('[:punct:]', ' '))) # replacing punctuat

  # for each element x in f_addr, splits into each character and sorts them, then collapses them; the e
  formatted_sorted <- as.character(sapply(formatted, function(x) {
    sorted_chars <- stri_sort(strsplit(x, NULL)[[1]])
    paste(sorted_chars, collapse = '')
    }))

  return(formatted_sorted)

}

# name normalizer
```

```
sorted_names = function(col) {
  # getting all clinics that are different but have the same name
  f_name_remove = c('remote', 'wic', 'field', 'office', 'mobile', 'clinic', ' ', '-')
  f_name_pattern = paste(f_name_remove, collapse = '|')
  f_name = str_replace_all(col, f_name_pattern, '')
  f_name = sorted(f_name)
  return(f_name)
}
```

Reformats

Add siteid column to `old`

turn everything lowercase

```
# old = old %>% mutate(date_open = ifelse(is.na(date_open), 'In 2009 but NA', date_open))

# extracting site id from `new`
rev_new_siteid = str_replace(str_extract(new$name, '\\d*-\\d*'), '-', '')
rev_new_siteid = str_replace(rev_new_siteid, '0*', '')

rev_new_name = str_replace(new$name, '\\d*-\\d*\\s*', '')

# creating new siteid column
new = new %>%
  mutate(siteid = rev_new_siteid, .before = name,
         name = rev_new_name)


# making everything lowercase for easy matching
old = old %>% mutate(COUNTY = str_to_lower(COUNTY),
                     name = str_to_lower(name),
                     state = str_to_lower(state),
                     city = str_to_lower(city))

# making county_name formatted the same
new = new %>% mutate(county_name = str_to_lower(
  str_replace(county_name, '\\sCounty\\s*', '')),
  name = str_to_lower(name),
  state = str_to_lower(state),
  city = str_to_lower(city))
```

make `f_addr` and `f_name`

```
new = new %>%
  mutate(f_addr = sorted(.$street_address), .after = street_address) %>%
  mutate(f_name = sorted_names(.$name), .after = name)

old = old %>%
  mutate(f_addr = sorted(.$address), .after = address) %>%
  mutate(f_name = sorted_names(.$name), .after = name)
```

```
old = old %>%
  mutate(
    date_open = ymd(date_open),
    date_close = ymd(date_close)
  )

old = old %>%
  mutate(
    addr_zip_county = paste0(f_addr, "_", zip, "_", COUNTY),
    name_zip_county = paste0(f_name, "_", zip, "_", COUNTY),
    siteid_zip = paste0(siteid, "_", zip)
  )

new = new %>%
  mutate(
    addr_zip_county = paste0(f_addr, "_", zipcode, "_", county_name),
    name_zip_county = paste0(f_name, "_", zipcode, "_", county_name),
    siteid_zip = paste0(siteid, "_", zipcode)
  )


old = old %>%
  mutate(
    fixed_date_open = date_open
  )
```

```
# fix the dates where both date_open and date_close are NA

old = old %>%
  mutate(
    fixed_date_open = case_when(
      is.na(date_open) & is.na(date_close) ~ as.Date("1993-01-01"),
      TRUE ~ fixed_date_open
    ),
    flag_both_dates_missing = is.na(date_open) & is.na(date_close)
  )
```

```
find_duplicates = function(clinic, all_clinics) {
# flow: for a given clinic, find other clinics where either f_addr & zip matches or f_name & zip matche
  potential_dups = all_clinics %>%
    filter(
      siteid != clinic$siteid, # not itself
      addr_zip_county == clinic$addr_zip_county | name_zip_county == clinic$name_zip_county
    )

  return(potential_dups)
}

# Find matches between new and old based on a key (siteid, name_zip_county, addr_zip_county)
find_matches = function(new_clinic, old_data, match_key) {
  if (match_key == "siteid_zip") {
    matches = old_data %>% filter(siteid_zip == new_clinic$siteid_zip)
  } else if (match_key == "name_zip_county") {
    matches = old_data %>% filter(name_zip_county == new_clinic$name_zip_county)
  } else if (match_key == "addr_zip_county") {
    matches = old_data %>% filter(addr_zip_county == new_clinic$addr_zip_county)
  } else {
    stop("Invalid match_key")
  }

  return(matches)
}




# Check if duplicates reopened within 31 days
check_reopening_31_days <- function(matches) {
  if (nrow(matches) < 2) return(FALSE)

  matches <- matches %>%
    arrange(date_open) # sort by date_open

  reopen_diffs <- difftime(matches$date_open[-1], matches$date_close[-nrow(matches)], units = "days")

  any(reopen_diffs >= 0 & reopen_diffs <= 31, na.rm = TRUE)
}



reopened_within_31_days = function(clinic, potential_dups) {
# Flow: for this one clinic `clinic`, find all potential duplicates and check if any of them reopened w
  if (nrow(potential_dups) == 0) {
    return(FALSE)
  }

# takes the duplicates and calculates the amount of time between their date_open and the closure date o
  potential_dups = potential_dups %>%
    mutate(days_diff = as.numeric(difftime(date_open, clinic$date_close, units = "days")))

# returns TRUE if any clinic's opening date was within 31 days of clinic i's closure date.
  any(potential_dups$days_diff >= 0 & potential_dups$days_diff <= 31, na.rm = TRUE)
}
```

```
# na_open_present_close = old %>%
#   filter(is.na(date_open) & !is.na(date_close))
#
#
# old = old %>%
#   mutate(
#     flag_open_na_close_present = FALSE,
#     flag_duplicate_found = FALSE,
#     flag_reopened_within_31 = FALSE
#   )
#
#
# for (i in 1:nrow(na_open_present_close)) {
#   clinic = na_open_present_close[i, ]
#
#
#   dups = find_duplicates(clinic, old)
#
#
#   old_idx = which(old$siteid == clinic$siteid)
#
#
#   old$flag_open_na_close_present[old_idx] = TRUE
#
#
#   old$flag_duplicate_found[old_idx] = nrow(dups) > 0
#
#
#   if (nrow(dups) > 0) {
#
#     reopened = reopened_within_31_days(clinic, dups)
#
#
#     old$flag_reopened_within_31[old_idx] = reopened
#
#
#     if (reopened) {
#
#       old$fixed_date_open[old_idx] = as.Date("1993-01-01")
#     } else {
#       dup_min_open = min(dups$date_open, na.rm = TRUE)
#
#       if (!is.na(dup_min_open) && clinic$date_close <= dup_min_open) {
#
#         old$fixed_date_open[old_idx] = as.Date("1993-01-01")
#       }
#     }
#   } else {
#
#     old$fixed_date_open[old_idx] = as.Date("1993-01-01")
#   }
# }
```

```
# 1. Initialize flag columns (run BEFORE the loop)
old = old %>%
  mutate(
    flag_open_na_close_present = FALSE,
    flag_duplicate_found = FALSE,
    flag_reopened_within_31 = FALSE
  )

# 2. Filter target clinics
na_open_present_close = old %>%
  filter(is.na(date_open) & !is.na(date_close))

# 3. Loop through each such clinic
for (i in 1:nrow(na_open_present_close)) {
  clinic = na_open_present_close[i, ]
  dups = find_duplicates(clinic, old)
  old_idx = which(old$siteid == clinic$siteid)
```

```r
  # Flag basic condition
  old$flag_open_na_close_present[old_idx] = TRUE
  old$flag_duplicate_found[old_idx] = nrow(dups) > 0

  if (nrow(dups) > 0) {
    # Check if any duplicate reopened within 31 days
    reopened = reopened_within_31_days(clinic, dups)
    old$flag_reopened_within_31[old_idx] = reopened

    # Check if any duplicate opened before this clinic closed
    opened_before_close = any(!is.na(dups$date_open) & dups$date_open < clinic$date_close)

    if (reopened && !opened_before_close) {
      # Case 1: Reopened quickly, and no earlier clinics - assume this is the original
      old$fixed_date_open[old_idx] = as.Date("1993-01-01")

    } else if (!reopened) {
      # Case 2: Not reopened - check for earliest known opening
      if (all(is.na(dups$date_open))) {
        # No known openings → assume this came first
        old$fixed_date_open[old_idx] = as.Date("1993-01-01")
      } else {
        dup_min_open = min(dups$date_open, na.rm = TRUE)
        if (clinic$date_close <= dup_min_open) {
          # Clinic closed before any other opened
          old$fixed_date_open[old_idx] = as.Date("1993-01-01")
        }
      }
    }
  } else {
    # Case 3: No duplicates - assume this is the first known clinic
    old$fixed_date_open[old_idx] = as.Date("1993-01-01")
  }
}
```

```r
# Find clinics in old where date_open is not NA but date_close is NA
open_present_close_na = old %>%
  filter(!is.na(date_open) & is.na(date_close))

# More flags
old = old %>%
  mutate(
    flag_open_present_close_na = FALSE,
    flag_open_present_close_na_duplicate_found = FALSE,
    flag_duplicate_came_first = NA,
    flag_fixed_due_to_earlier_duplicate = FALSE
  )

# Loop again through each clinic in open_present_close_na
for (i in 1:nrow(open_present_close_na)) {
  clinic = open_present_close_na[i, ]

  # Find potential duplicates for this clinic
  dups = find_duplicates(clinic, old)
  # Get this clinic's index in old
  old_idx = which(old$siteid == clinic$siteid)

  old$flag_open_present_close_na[old_idx] = TRUE
  old$flag_open_present_close_na_duplicate_found[old_idx] = nrow(dups) > 0

  # If at least one duplicate is found
  if (nrow(dups) > 0) {
    dups_with_earlier_open = dups %>%
      filter(
        # Filter duplicates with not NA date open and the date open is less than clinic i's date open
        (!is.na(date_open) & date_open < clinic$date_open) |
          # Filter duplicates with not NA date open and the date close is less than clinic i's date ope
        (!is.na(date_close) & date_close < clinic$date_open)
      )

    earlier_than_open = nrow(dups_with_earlier_open) > 0
    old$flag_duplicate_came_first[old_idx] = earlier_than_open

    if (earlier_than_open) {
      # Filter duplicates that came before clinic i that have NA date open
      dups_needing_fix = dups_with_earlier_open %>%
        filter(is.na(date_open))

      if (nrow(dups_needing_fix) > 0) {
        old = old %>%
          mutate(
            # Change these date opens to 1993
            fixed_date_open = if_else(
              siteid %in% dups_needing_fix$siteid,
              as.Date("1993-01-01"),
              fixed_date_open
            ),
            flag_fixed_due_to_earlier_duplicate = if_else(
              siteid %in% dups_needing_fix$siteid,
              TRUE,
              flag_fixed_due_to_earlier_duplicate
            )
          )
      }
    }
  }
}
```

```
old = old %>%
  mutate(fixed_date_open = as.Date(fixed_date_open))
```

```
match_new_to_old = function(new_data, old_data) {

  results = list()

  for (i in 1:nrow(new_data)) {
    new_clinic = new_data[i, ]
    matched_by_id = FALSE
    matched_by_name = FALSE
    matched_by_addr = FALSE

    matched_clinics = tibble()

    # Check each matching method separately
    matches_id = find_matches(new_clinic, old_data, "siteid_zip")
    if (nrow(matches_id) > 0) {
      matched_by_id = TRUE
      matches_id = matches_id %>% mutate(match_method = "siteid_zip")
      matched_clinics = bind_rows(matched_clinics, matches_id)
    }

    matches_name = find_matches(new_clinic, old_data, "name_zip_county")
    if (nrow(matches_name) > 0) {
      matched_by_name = TRUE
      matches_name = matches_name %>% mutate(match_method = "name_zip_county")
      matched_clinics = bind_rows(matched_clinics, matches_name)
    }

    matches_addr = find_matches(new_clinic, old_data, "addr_zip_county")
    if (nrow(matches_addr) > 0) {
      matched_by_addr = TRUE
      matches_addr = matches_addr %>% mutate(match_method = "addr_zip_county")
      matched_clinics = bind_rows(matched_clinics, matches_addr)
    }

    if (nrow(matched_clinics) == 0) {
      results[[i]] = new_clinic %>%
        mutate(
          by_id = FALSE,
          by_name = FALSE,
          by_addr = FALSE,
          match_found = FALSE,
          flag_multiple_matches = FALSE,
          flag_missing_date = FALSE,
          flag_tie_open = FALSE,
          fixed_siteid = NA,
          fixed_date_open = NA,
          within_31_days = NA,
          max_date = NA
        )
    } else {
      match_found = TRUE

      missing_date = any(is.na(matched_clinics$fixed_date_open))
      reopening_within_31 = check_reopening_31_days(matched_clinics)

      if (reopening_within_31) {
        within_31_days = TRUE
        earliest_date = min(matched_clinics$fixed_date_open, na.rm = TRUE)
        matches_earliest = matched_clinics %>% filter(fixed_date_open == earliest_date)
```

```
        tie_open = nrow(matches_earliest) > 1
        fixed_siteid = paste(matches_earliest$siteid, collapse = ";")
        fixed_date_open = earliest_date
        flag_multiple_matches = tie_open
        max_date = FALSE
      } else {
        within_31_days = FALSE
        fixed_siteid = paste(matched_clinics$siteid, collapse = ";")

        valid_dates = matched_clinics %>% filter(!is.na(fixed_date_open))

        if (nrow(valid_dates) == 0) {
          max_date = NA
          fixed_date_open = NA
        } else if (any(matched_clinics$flag_reopened_within_31 == TRUE)) {
          t_reopen = matched_clinics %>% filter(flag_reopened_within_31 == TRUE)
          if (all(is.na(t_reopen$fixed_date_open))) {
            fixed_date_open = NA
          } else {
            fixed_date_open = min(t_reopen$fixed_date_open, na.rm = TRUE)
          }
          max_date = FALSE
        } else {
          max_date = TRUE
          fixed_date_open = max(valid_dates$fixed_date_open)
        }

        flag_multiple_matches = TRUE
        tie_open = NA
      }

      results[[i]] = new_clinic %>%
        mutate(
          by_id = matched_by_id,
          by_name = matched_by_name,
          by_addr = matched_by_addr,
          match_found = match_found,
          within_31_days = within_31_days,
          flag_multiple_matches = flag_multiple_matches,
          flag_missing_date = missing_date,
          flag_tie_open = tie_open,
          fixed_siteid = fixed_siteid,
          fixed_date_open = fixed_date_open,
          max_date = max_date
        )
    }
  }

  final_results = bind_rows(results)
  return(final_results)
}

matched_new = match_new_to_old(new, old)
```

```r
vec_names = names(matched_new)[-c(3, 5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22)]
matched_new = matched_new %>% select(all_of(vec_names), siteid_zip, name_zip_county, addr_zip_county, wi

matched_new %>% filter(fixed_date_open == Inf)
```

```
## # A tibble: 0 x 21
## # i 21 variables: siteid <chr>, name <chr>, street_address <chr>, city <chr>,
## #   state <chr>, zipcode <chr>, county_name <chr>, by_id <lgl>, by_name <lgl>,
## #   by_addr <lgl>, match_found <lgl>, within_31_days <lgl>,
## #   flag_multiple_matches <lgl>, flag_missing_date <lgl>, flag_tie_open <lgl>,
## #   fixed_siteid <chr>, fixed_date_open <date>, max_date <lgl>,
## #   siteid_zip <chr>, name_zip_county <chr>, addr_zip_county <chr>
```

```r
nrow(matched_new %>% filter(!is.na(fixed_date_open))) # Found 376 dates (now 373)
```

```
## [1] 373
```

```r
nrow(matched_new %>% filter(is.na(fixed_date_open))) # Missing 89 dates (now 92)
```

```
## [1] 92
```

```r
together = list()

NA_matched_new = matched_new %>% filter(is.na(fixed_date_open))
NA_matched_new_siteid = NA_matched_new %>% pull(siteid)

NA_matched_new_in_old = old %>% filter(siteid %in% NA_matched_new_siteid)
nrow(NA_matched_new_in_old) # 33
```

```
## [1] 35
```

```r
for (i in 1:nrow(NA_matched_new_in_old)) {
  clinic = NA_matched_new_in_old[i, ]
  id = clinic %>% pull(siteid)

  clinic_in_new = new %>% filter(siteid == id) %>% mutate(from = "new")
  clinic_in_old = old %>% filter(siteid == id) %>% mutate(from = "old")

  dups = find_duplicates(clinic, old) %>% mutate(from = "duplicate in old")

  together = bind_rows(together, bind_rows(clinic_in_new, clinic_in_old, dups))


}

together = together %>%
  mutate(zipcode = coalesce(zipcode, zip)) %>%
  select(siteid, name, f_name, f_addr, city, zipcode, fixed_date_open, date_close, from)
```

```r
NA_with_old = NA_matched_new %>%
  left_join(NA_matched_new_in_old, by = c("siteid" = "siteid", "city" = "city"), suffix = c(".new", ".ol

matched_fix = NA_with_old %>%
  filter(!is.na(fixed_date_open.old)) %>%
  select(siteid, fixed_date_open.old)
```

```
matched_new_it2 = matched_new %>%
  left_join(matched_fix, by = "siteid") %>%
  mutate(fixed_date_open = if_else(is.na(fixed_date_open) & !is.na(fixed_date_open.old), fixed_date_open
         bad_zipcode = if_else(!is.na(fixed_date_open.old), TRUE, FALSE)) %>%
  select(-fixed_date_open.old)



nrow(matched_new_it2 %>% filter(is.na(fixed_date_open))) # Missing 61 dates (now 63)
```

```
## [1] 63
```

```
# checked the extra two and it's good (vidor clinics)
```

```
remaining = matched_new_it2 %>% filter(is.na(fixed_date_open)) %>% pull(siteid)
remaining_in_old = old %>% filter(siteid %in% remaining)

# There are 5 clinics in matched_new_it2 that exist in old as well.
nrow(remaining_in_old)
```

```
## [1] 6
```

```
remaining_ids = remaining_in_old %>% pull(siteid)
remaining_in_matched = matched_new_it2 %>% filter(siteid %in% remaining_ids)

check = bind_rows(remaining_in_matched, remaining_in_old) %>% arrange(siteid)

# I manually checked these 5 (now 6) clinics. They appear to have moved cities. Therefore, will remain
```

```
matched_finalized = matched_new_it2 %>% select(-c(flag_tie_open, fixed_siteid, flag_multiple_matches))

# Note that if bad_zipcode is TRUE then the previous fixed_date_open would've been NA
```

```
sam = read_csv("/Users/teorichard/Downloads/UCD Research/Texas_WIC_Research_Files/MergedMatchTeoSam.csv
```

```
## Rows: 465 Columns: 26
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): name, address, city, phone, email, date_open, street_address, sta...
## dbl   (7): siteid, zip, sample_date, BY_SITEID, BY_ADDRESS, BY_NAME, zipcode
## lgl   (8): COUNTY, date_close, by_id, by_name, by_addr, match_found, flag_mi...
## date  (1): fixed_date_open
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sam = sam %>% mutate(siteid = as.character(siteid))

sam %>% select(siteid, name, address, zip, date_open, fixed_date_open) %>%
  filter(date_open != fixed_date_open)
```

```
## # A tibble: 21 x 6
##    siteid name                        address   zip date_open fixed_date_open
##    <chr>  <chr>                       <chr>   <dbl> <chr>     <date>
##  1 129    elgin                       218 s ~ 78621 2009-02-~ 1994-01-01
##  2 1217   mission ii                  722 n ~ 78572 1994-05-~ 1993-01-01
##  3 1218   mcallen ii                  220 s ~ 78501 1995-04-~ 1993-01-01
##  4 1224   pharr ii                       300 w ~ 78577 1999-01--~ 1993-01-01
##  5 1228   mobile unidos podemos          3105 w~ 78539 2002-06--~ 1996-07-22
##  6 3001   city of port arthur health de~ 5860 9~ 77642 1993-01--~ 1999-08-10
##  7 3908   daingerfield                   1402 b~ 75638 2010-01--~ 1996-12-01
##  8 3917   jefferson                      1113 n~ 75657 1998-11--~ 1993-01-01
##  9 3919   carthage                       446 w ~ 75633 1998-11--~ 1993-01-01
## 10 4808   southeast center               3737 r~ 77503 1996-08--~ 1993-01-01
## # i 11 more rows
```

```r
names(sam)
```

```
##  [1] "siteid"            "name"              "address"
##  [4] "city"              "zip"               "COUNTY"
##  [7] "date_close"        "sample_date"       "phone"
## [10] "email"             "BY_SITEID"         "BY_ADDRESS"
## [13] "BY_NAME"           "date_open"         "street_address"
## [16] "state"             "zipcode"           "county_name"
## [19] "by_id"             "by_name"           "by_addr"
## [22] "match_found"       "flag_missing_date" "fixed_date_open"
## [25] "bad_zipcode"       "_merge"
```

```r
left_join(sam, matched_finalized, by = "siteid", suffix = c("", ".y")) %>%
  mutate(fixed_date_open = fixed_date_open.y) %>%
  select(names(sam)) %>% select(siteid, name, street_address, zip, date_open, fixed_date_open)
```

```
## # A tibble: 465 x 6
##    siteid name                    street_address   zip date_open fixed_date_open
##    <chr>  <chr>                   <chr>          <dbl> <chr>     <date>
##  1 104    st johns community cen~ 7500 Blessing~ 78752 1993-01--~ 1993-01-01
##  2 105    northwest               8701 Research~ 78758 1993-11--~ 1993-11-15
##  3 107    montopolis neighborhoo~ 2901 Montopol~ 78741 1993-01--~ 1993-01-01
##  4 109    far south               405 W Stassne~ 78745 1993-01--~ 1993-01-01
##  5 112    dove springs            5811 Palo Bla~ 78744 1993-03--~ 1993-03-15
##  6 114    manor                   14008 Shadowg~ 78653 1993-01--~ 1993-01-01
##  7 115    pflugerville            15822-B Footh~ 78660 1993-01--~ 1993-01-01
##  8 121    del valle               3518 FM 973    78617 1993-01--~ 1993-01-01
##  9 128    bastrop                 605 Old Austi~ 78602 2009-02--~ 2009-02-02
## 10 129    elgin                   218 South Mai~ 78621 2009-02--~ 1994-01-01
## # i 455 more rows
```

```r
nomatch_NA = matched_finalized %>% filter(is.na(fixed_date_open))
```

```r
manual_fixes = tibble(
  siteid = c("7717", "9003", "13126", "13195", "13306", "6411", "6902", "5110", "5111", "332",
             "4210", "13172", "13161", "11002", "13020", "13025", "13124", "13018", "13115",
             "13151", "13030", "13041", "3317"),
  fixed_date_open = as.Date(c("2006-10-01", "1994-10-01", "1993-09-01",
                              "1998-01-01", "1993-01-01", "1997-02-03",
                              "1993-01-01", "1993-01-01", "1995-04-04",
                              "1993-01-01", "1993-01-01", "1998-01-15",
                              "1993-01-01", "2005-10-01", "1994-05-01",
                              "1994-05-01", "1993-09-01", "1994-10-05",
                              "1993-01-01", "1995-05-03", "1996-10-01",
                              "1996-01-01", "1995-02-01"

                              ))
```

```
) %>% mutate(
  manual_date_change = TRUE
)

leave_NA = tibble(
  siteid = c("3925", "742", "743", "13307", "6110", "13153", "13197",
             "13198", "2209", "7723", "2908", "2901", "4602", "1112",
             "6307", "13308", "13305", "6410", "8964", "5935", "5914",
             "8963", "5915", "1318", "1322", "1320", "8965", "8967",
             "4305", "13029", "13021", "13024", "13123", "13005", "13009",
             "13008", "13002", "13004", "13010", "13003"
             ),
  why_NA = c("city mismatch", "not found", "not found", "not found", "not found",
             "not found", "not found", "city mismatch", "not found", "not found",
             "county mismatch", "not found", "city mismatch", "not found",
             "FOUND, NA date", "not found", "not found", "not found", "not found",
             "not found", "not found", "not found", "not found", "not found",
             "not found", "not found", "city mismatch", "not found", "not found",
             "not found", "not found", "not found", "not found", "city mismatch",
             "not found", "not found", "not found", "not found", "not found",
             "not found"
             )
)

manual_bad_zip = tibble(
  siteid = c("6411", "7717", "9003", "332", "4210", "13018", "13030", "3317"),
  bad_zipcode = TRUE
)

manual_matched_finalized = left_join(matched_finalized, manual_fixes, by = "siteid", suffix = c("", ".y
  mutate(fixed_date_open = coalesce(fixed_date_open, fixed_date_open.y)) %>%
  left_join(leave_NA, by = "siteid") %>%
  left_join(manual_bad_zip, by = "siteid", suffix = c("", ".y")) %>%
  mutate(bad_zipcode = case_when(
    bad_zipcode == FALSE & bad_zipcode.y == TRUE ~ TRUE,
    bad_zipcode == TRUE & is.na(bad_zipcode.y) ~ TRUE,
    TRUE ~ FALSE
  )) %>%
  select(all_of(names(matched_finalized)), manual_date_change, why_NA)

manual_matched_finalized %>% filter(is.na(fixed_date_open))
```

```
## # A tibble: 40 x 21
##    siteid name      street_address city  state zipcode county_name by_id by_name
##    <chr>  <chr>     <chr>          <chr> <chr> <chr>   <chr>       <lgl> <lgl>
##  1 3925   gun barr~ 1901 W Main St gun ~ tx    75156   henderson   FALSE FALSE
##  2 742    fruitdal~ 4408 Vandervo~ dall~ tx    75216   dallas      FALSE FALSE
##  3 743    healing ~ 5750 Pineland~ dall~ tx    75231   dallas      FALSE FALSE
##  4 13307  buffalo ~ 942 North Hil~ buff~ tx    75831   leon        FALSE FALSE
##  5 6110   kirbyvil~ 204 MLK Ave    kirb~ tx    75956   newton      FALSE FALSE
##  6 13153  nocona w~ Community Cen~ noco~ tx    76255   montague    FALSE FALSE
##  7 13197  comanche~ 209 W Duncan ~ coma~ tx    76442   comanche    FALSE FALSE
##  8 13198  de leon ~ Old Hotel Apa~ de l~ tx    76444   comanche    FALSE FALSE
##  9 2209   south 18~ 1800 Gurley Ln waco  tx    76706   mclennan    FALSE FALSE
## 10 7723   lone sta~ 605 S Conroe ~ conr~ tx    77304   montgomery  FALSE FALSE
## # i 30 more rows
## # i 12 more variables: by_addr <lgl>, match_found <lgl>, within_31_days <lgl>,
## #   flag_missing_date <lgl>, fixed_date_open <date>, max_date <lgl>,
## #   siteid_zip <chr>, name_zip_county <chr>, addr_zip_county <chr>,
## #   bad_zipcode <lgl>, manual_date_change <lgl>, why_NA <chr>
```

```
# knitr::purl("texas_merge_try_again.Rmd", output = "texas_merge_setup.R")
```